|  |
| --- |
| createMax:  size = 1000, min = 17.0000, average = 22.0400, median = 22.0000, sd = 1.8255, max = 30.0000 |
| createAverage:  size = 1000, min = 10.6579, average = 12.0050, median = 11.9091, sd = 0.6313, max = 15.2796 |
| deleteMax:  size = 1000, min = 15.0000, average = 19.3800, median = 19.0000, sd = 1.7611, max = 26.0000 |
| deleteAverage:  size = 1000, min = 9.4434, average = 10.6392, median = 10.5410, sd = 0.6207, max = 14.1348 |
| reinsertMax:  size = 1000, min = 16.0000, average = 21.0820, median = 21.0000, sd = 2.0123, max = 30.0000 |
| reinsertAverage:  size = 1000, min = 9.9817, average = 11.3075, median = 11.2021, sd = 0.6671, max = 14.7281 |

Project 3 Report

In this simulation we find out that although the BST that we create is valid its not the best BST that we can create because due to the fact that the data is randomly shuffled it doesn’t create a perfectly balanced BST it does create some parts of the tree that are balanced so the BST is still faster to find information in than linked list but we still have the possibility of having an unbalanced tree so if we look at our data with the power of 10 meaning if the tree is perfectly balanced we should get 10 levels but we find that the max is between 17 and 30 levels meaning that we are off from the perfectly balanced tree by 7 to 10 levels and but I think that that is okay because that only the max values but when you look at the average values that we got we see that its much closer to 10 ranging between 10.6579 to 15.2796 which still shows that we are not getting the best possible tree but we are close and that difference gets closer when we delete and then reinsert because the tree shuffles some of the elements into there correct places